



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,634	10/01/2003	Luis M. Gomes	5150-82801	7873

7590 02/07/2007
Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER

AUGUSTINE, NICHOLAS

ART UNIT	PAPER NUMBER
----------	--------------

2179

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary

Application No.

10/676,634

Applicant(s)

GOMES ET AL.

Examiner

Nicholas Augustine

Art Unit

2179

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12/02/2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-23 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-23 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
- Paper No(s)/Mail Date 8/19/2004.

- 4) ☐ Interview Summary (PTO-413)
- Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1 and 18-21 are rejected under 35 U.S.C. 103(a) as being obvious over PerfectXML in view of Michelman et al (US 2002/0075309 A1).

As to independent claims 1 and 18-21, Michelman teaches a memory medium and corresponding system and method which stores program instructions implementing a graphical user interface (GUI) for debugging a program, wherein, during execution of the program, the program instructions are executable by a processor to perform (PerfectXML, Page 116, Paragraph 3): displaying source code for the program in a first GUI element (PerfectXML, Fig. 7.3); PerfectXML does not specifically mention that the tool tip accepts user input, however in the same field of endeavor Michelman teaches receiving first user input to the first GUI element indicating an expression in the source code (col.2, lines 66-67 and 1-3; wherein the user places a cursor over an object such that of an expression in a debug application); displaying a value of the expression in a tool tip in response to said first user input (col.4, line 29; displaying information); Michelman teaches receiving second user input to the tool tip modifying the displayed value (col.4, line 31; wherein the user is able to edit, within a tool tip, the information being displayed as a result of cursor placement within in an application that of debugging for example), thereby specifying a new value for the expression (col. 4, line 31; wherein the edited information is new upon user interaction with the tool tip (col.9, line 21-24)); and PerfectXML teaches setting the expression in the program to the new value, wherein the program continues execution in accordance with the new value of the expression (PerfectXML, pg. 108). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET). It is appreciated in the art that

Microsoft Visual Studio debugger application as taught by PerfectXML is used to correct expressions and the like using an input means supplemental window pane called the watch window. This watch window updates information and allows the user to interact with the displayed source code, the user can mouse over source code expressions and a tool tip will appear with the values that of displayed and editable in the watch window, the watch window acts as a medium of interaction and therefore would have been obvious to combine the system of Michelman into the system of PerfectXML because the system and method can be modified to address a variety of other scenarios which that of the watch window replacement (col.7 , line 25).

Note: that the main inventive idea is that of an editable tool tip which is provided and taught by Michelman whom suggest the use of his invention in any system and method to match any scenario (col.7, line 10 and 25). Those skilled in the art will appreciate that the combination of the two mentioned above references shows that of the main inventive idea being used for the scenario of the product Microsoft Visual Studio Debugge

6. Claims 1-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over PerfectXML (<http://www.perfectxml.com/nr/aspnetdebug.pdf>) in view of East Tennessee State University (http://csciwww.etsu.edu/blair/Using_Debugger.htm).

Note: that the references both teach the same debugging environment from the same development software suite Microsoft Visual Studio .NET, but for purposes of

Art Unit: 2179

discussion both references are listed above where PerfectXML teaches the main concept of using a debugger in Visual Studio.NET and East Tennessee State University (referred to as ETSU herein) discloses with figures when the user hovers over a variable(s) in the source code a tool tip will be displayed within the use of the debugger.

Also some content pertaining to the immediate application is not depicted in these articles because of it's commonness in the art.

As to independent claims 1 and 18-21, PerfectXML teaches a memory medium and corresponding system and method which stores program instructions implementing a graphical user interface (GUI) for debugging a program, wherein, during execution of the program, the program instructions are executable by a processor to perform (PerfectXML, Page 116, Paragraph 3): displaying source code for the program in a first GUI element (PerfectXML, Fig. 7.3); PerfectXML teaches receiving second user input **to the tool tip** modifying the displayed value (PerfectXML, pg. 108, par. 3), thereby specifying a new value for the expression (PerfectXML, pg. 108, par. 3); and setting the expression in the program to the new value, wherein the program continues execution in accordance with the new value of the expression (PerfectXML, pg. 108). **Of course those skilled in the art will appreciate that the watch window is accepting user interaction of modifying a displayed expression in the displayed area called a tool tip, this watch window is an editable means "to the tool tip", wherein the tool tip being that of a display area to display information from the watch window.**

Art Unit: 2179

PerfectXML does not specifically mention receiving first user input to indicate an expression; displaying a value in a tool tip; wherein the window does not include window title bars or menus; however in the same field of endeavor ETSU teaches receiving first user input to the first GUI element indicating an expression in the source code (ETSU, pg. 7); displaying a value of the expression in a tool tip in response to said first user input (ETSU, pg. 7); wherein the window does not include window title bars or menus (ETSU, page 8); It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because PerfectXML and ETSU disclose a brief introduction of the Microsoft Visual C++ debugger, being that of an introduction not all detailed functionality was disclosed thus the combination of references was used because the brief introduction of PerfectXML did not specifically mention a feature/ functionality of the Microsoft Visual C++ debugger that ETSU did disclose in detail (page 1 of ETSU). Also the motivation to combine is the presence of the watch window which is disclosed in PerfectXML (figure 7.4) and ETSU (page 9), wherein PerfectXML shows the functionality and touches briefly on what ETSU covers in more detail on the functionality as mentioned above.

As to claims 2-17 and 22-23, note the discussion above PerfectXML in view of ETSU, with respect to their teaching of the SAME debugging environment from SAME software product (Microsoft Visual Studio .NET).

As to dependent claim 2, PerfectXML in view of ETSU teaches the memory medium of claim 1, PerfectXML does not specifically mention receiving input however in the same field of endeavor ETSU teaches wherein said receiving first user input to the first GUI element comprises: receiving the first user input from a pointing device (ETSU, page 7 (mouse)). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 3, PerfectXML in view of ETSU teaches the memory medium of claim 2, PerfectXML does not specifically mention a cursor, hovering however in the same field of endeavor ETSU teaches wherein the first user input from a pointing device comprises: a cursor associated with the pointing device hovering over the expression (ETSU, page 7) (Although not depicted in this figure, because of the limitations of the 'print screen' function on a keyboard of a PC. It is known in the art that a cursor is associated with a mouse). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 4, PerfectXML in view of ETSU teaches the memory medium of claim 1, wherein said wherein said receiving first user input to the first GUI element comprises: receiving the first user input from a keyboard. (PerfectXML, pg. 108) (It is inherent to one of ordinary skill in the art that a development software product makes use of user input through means of a combination of a keyboard, mouse, menus, or the like).

As to dependent claim 5, PerfectXML in view of ETSU teaches the memory medium of claim 1, wherein said wherein said receiving first user input to the first GUI element comprises: receiving the first user input to a menu. (PerfectXML, pg. 108) (It is inherent to one of ordinary skill in the art that a development software product makes use of user input through means of a combination of a keyboard, mouse, menus, or the like).

As to dependent claim 6, PerfectXML in view of ETSU teaches the memory medium of claim 1, PerfectXML does not specifically mention context sensitive however in the same field of endeavor ETSU teaches wherein the tool tip is context sensitive (ETSU, page 7) (It is known in the art that defined variables are context sensitive and since the tool tips context is made up defined variable(s) it as well would be context sensitive). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both

Art Unit: 2179

disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 7, PerfectXML in view of ETSU teaches the memory medium of claim 6, PerfectXML does not specifically mention a control, type data however in the same field of endeavor ETSU teaches wherein the tool tip comprises a control corresponding to a data type of the expression, and wherein the data type of the expression comprises at least one of:

A string data type; a character data type (ETSU, page 7); a numeric data type (ETSU, page 7); a Boolean data type; and an array data type. (It is recognized and well practiced in the art that programming makes use of string data type, Boolean data type and array data type. It is apparent by the visual representation of the tool tips depicted in the figures of ETSU that they will display any value assigned to an expression or variables in any typical common format.) It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 8, PerfectXML in view of ETSU teaches the memory medium of claim 6, PerfectXML does not specifically mention format however in the same field of

Art Unit: 2179

endeavor ETSU teaches wherein the tool tip is operable to display the value of the expression in a specified format; wherein if the expression comprises integer data, the specified format comprises one or more of: decimal (ETSU, page 7); hexadecimal (ETSU, page 7); octal; binary; and ASCII (ETSU, page 7); and wherein if the expression comprises single or double precision, the specified format comprises one or more of floating point; and scientific notation. (It is recognized and well practiced in the art that programming makes use of octal, binary, single or double precision, one or more of floating point numbers and scientific notation; it is apparent by the visual representation of the tool tips depicted in the figures of ETSU that they will display any value assigned to an expression or variables in any typical common format.) It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 9, PerfectXML in view of ETSU the memory medium of claim 8, wherein the specified format is specified via a second GUI element in the GUI (PerfectXML, Fig. 7.4, (Type))

As to dependent claim 10, PerfectXML in view of ETSU the memory medium of claim 1, PerfectXML does not specifically mention display portions however in the same field of

Art Unit: 2179

endeavor ETSU teaches wherein the tool tip comprises: a first portion, operable to display the value of the expression, wherein the first portion is further operable to receive the second user input modifying the value; and a second portion, operable to display non-editable information related to the expression (ETSU, page 7). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 11, PerfectXML in view of ETSU the memory medium of claim 10, PerfectXML does not specifically mention second portion however in the same field of endeavor ETSU teaches wherein the second portion comprises a text indicator, operable to display text (ETSU, page 7). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 12, PerfectXML in view of ETSU the memory medium of claim 10, PerfectXML does not specifically mention first portion however in the same field of endeavor ETSU teaches wherein the first portion is further operable to graphically

Art Unit: 2179

indicate that the value is editable (ETSU, page 7; it is known in the art that editable fields show a blinking cursor awaiting input, so when the user clicks in the field the flashing cursor is presented). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 13, PerfectXML in view of ETSU the memory medium of claim 1, PerfectXML does not specifically mention wherein expression value however in the same field of endeavor ETSU teaches the expression comprises a variable (ETSU, page 8). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 14, PerfectXML in view of ETSU the memory medium of claim 1, PerfectXML does not specifically mention wherein syntactic expression however in the same field of endeavor ETSU teaches the expression comprises a syntactic expression comprising one or more of: one or more variables (ETSU, page 8 (nStackTop, nValue, nStackArray)); one or more constants; one or more macros; and one or more operators

Art Unit: 2179

(ETSU, page 8). (For purposes of lack of description in figures and content of ETSU and PerfectXML, it is well known in the art that an expression will comprise one or more of variables, constants, macros, or operators.) It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 15, PerfectXML in view of ETSU the memory medium of claim 1, PerfectXML does not specifically mention debugging mode however in the same field of endeavor ETSU teaches wherein the execution of the program is in debugging mode (ETSU, page 1 and 7; (debug controls running the program are depicted in figures). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 16, PerfectXML in view of ETSU the memory medium of claim 1, PerfectXML does not specifically mention instructions to perform function however in the same field of endeavor ETSU teaches wherein the program instructions are further executable to perform; evaluating the expression to determine the value of the

Art Unit: 2179

expression (ETSU, page 8; (nStackTop = 1 from the expression '++nStackTop')). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 17, PerfectXML in view of ETSU the memory medium of claim 1, wherein the program instructions are further executable to perform: dismissing the tool tip based on one or more of: third user input, indicating dismissal of the tool tip (It is well known in the art that when the user takes the mouse away from the object/link (cursor is no longer hovering over) that the tool tip goes away); and lapse of a specified time period (Although not depicted in this article because of it's commonness it is well known in the art to have the tool tip dismiss itself with a set time, see Microsoft Windows XP operating system for an example, when the user hovers over a tab in the task bar the tool tip is shown). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

Art Unit: 2179

As to dependent claim 22, PerfectXML does not specifically mention size however in the same field of endeavor ETSU teaches wherein the window is substantially just large enough to display the value of the indicated expression (ETSU, page 8 (nStackTop =1)). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

As to dependent claim 23, PerfectXML does not specifically mention display in the window however in the same field of endeavor ETSU teaches wherein the window is further operable to display the indicated expression, and wherein the program instructions are further executable to perform: displaying the indicated expression with the value in the window, wherein the window does not include visible boundaries demarcating the displayed expression and value, wherein the window is substantially just large enough to display the indicated expression and the value of the indicated expression (ETSU, page 8). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of ETSU into the teachings of PerfectXML, because they both disclose the debugging environment of the same software application (Microsoft Visual Studio .NET).

It is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006,1009, 158 USPQ 275, 277 (CCPA 1968)).

Conclusion

- Introduction to Visual C++ Debugging -
(http://ei.cs.vt.edu/~cs1205/c_debug/intro.html) an overview guide to features within the environment.
- Microsoft Word 2000 – See included figures, which depict a tool tip that accepts user input.
- InfoWorld News – “Gates Casts Visual Studio .Net”
(<http://www.infoworld.com/articles/hn/xml/02/02/13/020213hngates2.html>) showing a date of arrival.

Inquires

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nicholas Augustine whose telephone number is 571-270-1056. The examiner can normally be reached on Monday - Friday: 7:30- 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Art Unit: 2179

Nicholas Augustine
Examiner
AU 2179

N. Augustine
1-23-2006

BA HUYNH
PRIMARY EXAMINER